WHAT IS CLAIMED IS:

1. A system for providing a floating point square root, comprising:

an analyzer circuit configured to determine a first status of a first floating point operand based upon data within the first floating point operand; and

a results circuit coupled to the analyzer circuit and configured to assert a resulting floating point operand containing the square root of the first floating point operand and a resulting status embedded within the resulting floating point operand.

2. The system for providing a floating point square root of claim 1, wherein the analyzer circuit further comprises:

a first operand buffer configured to store the first floating point operand; and

a first operand analysis circuit coupled to the first operand buffer, the first operand analysis circuit configured to generate a characteristic signal having information relating to the first status, the characteristic signal comprising a portion of the at least one control signal.

3. The system for providing a floating point square root of claim 2, wherein the first status is determined without regard to memory storage external to the first operand buffer.

4. The system for providing a floating point square root of claim 3, wherein the memory storage external to the first operand buffer is a floating point status register.

5. The system for providing a floating point square root of claim 1, wherein the results circuit further comprises:

a square root circuit coupled to the analyzer circuit, the square root circuit configured to produce the square root of the first floating point operand;

a square root logic circuit coupled to the analyzer circuit and configured to produce the resulting status based upon the first status; and

a result assembler coupled to the square root circuit and the square root logic circuit, the result assembler configured to assert the resulting floating point operand and embed the resulting status within the resulting floating point operand.

6. The system for providing a floating point square root of claim 5, wherein the square root logic circuit is organized according to the structure of a decision table.

7. The system for providing a floating point square root of claim 1, wherein the first status and the resulting status are each one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

8. The system for providing a floating point square root of claim 7, wherein the overflow status represents one in a group of a +OV status and a -OV status.

9. The system for providing a floating point square root of claim 8, wherein the overflow status is represented as a predetermined non-infinity numerical value.

10. The system for providing a floating point square root of claim 7, wherein the underflow status represents one in a group of a +UN status and a -UN status.

11. The system for providing a floating point square root of claim 10, wherein the underflow status is represented as a predetermined non-zero numerical value.

12. The system for providing a floating point square root of claim 7, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

13. The system for providing a floating point square root of claim 7, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.

14. A method for providing a floating point square root, comprising:

determining a first status of a first floating point operand based upon data within the first floating point operand; and

asserting a resulting floating point operand containing the square root of the first floating point operand and a resulting status embedded within the resulting floating point operand.

15. The method for providing a floating point square root of claim 14, wherein the determining stage further comprises:

storing the first floating point operand in a first operand buffer; and

generating a first characteristic signal representative of the first status.

16. The method for providing a floating point square root of claim 15, wherein the first status is determined without regard to memory storage external to the first operand buffer.

17. The method for providing a floating point square root of claim 16, wherein the memory storage external to the first operand buffer is a floating point status register.

18. The method for providing a floating point square root of claim 14, wherein the asserting stage further comprises:

producing the square root of the first floating point operand; and

asserting the resulting floating point operand.

19. The method for providing a floating point square root of claim 14, wherein at least one of the following: the first status, and the resulting status comprise at least one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

20. The method for providing a floating point square root of claim 19, wherein the overflow status represents one in a group of a +OV status and a -OV status.

21. The method for providing a floating point square root of claim 20, wherein the overflow status is represented as a predetermined non-infinity numerical value.

22. The method for providing a floating point square root of claim 19, wherein the underflow status represents one in a group of a +UN status and a -UN status.

23. The method for providing a floating point square root of claim 22, wherein the underflow status is represented as a predetermined non-zero numerical value.

24. The method for providing a floating point square root of claim 19, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

25. The method for providing a floating point square root of claim 19, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.

26. A computer-readable medium on which is stored a set of instructions for providing a floating point square root, which when executed perform stages comprising:

determining a first status of a first floating point operand based upon data within the first floating point operand; and

asserting a resulting floating point operand containing the square root of the first floating point operand and a resulting status embedded within the resulting floating point operand.

27. The computer-readable medium of claim 26, wherein the determining stage further comprises:

storing the first floating point operand in a first operand buffer; and

generating a second characteristic signal representative of the first status.

28. The computer-readable medium of claim 27, wherein the first status is determined without regard to memory storage external to the first operand buffer.

29. The computer-readable medium of claim 28, wherein the memory storage external to the first operand buffer is a floating point status register.

30. The computer-readable medium of claim 26, wherein the asserting stage further comprises:

producing the square root of the first floating point operand; and

asserting the resulting floating point operand.

31. The computer-readable medium of claim 26, wherein at least one of the following: the first status, and the resulting status comprise at least one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

32. The computer-readable medium of claim 31, wherein the overflow status represents one in a group of a +OV status and a -OV status.

33. The computer-readable medium of claim 32, wherein the overflow status is represented as a predetermined non-infinity numerical value.

34. The computer-readable medium of claim 31, wherein the underflow status represents one in a group of a +UN status and a -UN status.

35. The computer-readable medium of claim 34, wherein the underflow status is represented as a predetermined non-zero numerical value.

36. The computer-readable medium of claim 31, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

37. The computer-readable medium of claim 31, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.